

REMARKS

This application has been reviewed in light of the Office action dated March 7, 2007. Claims 1-28 are pending in the application. No new matter has been added. The Examiner's reconsideration of the rejection in view of the following remarks is respectfully requested.

The Applicant acknowledges with appreciation the telephone interview granted by the Examiner on June 6, 2007. The undersigned and the Examiner discussed the relevance of Hasegawa in view of the current specification and claims. Agreement was not reached on the relevance of Hasegawa.

By the Office Action, claims 1-28 stand rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the applicant regards as the invention.

The Applicant respectfully disagrees with the rejections.

“Should be” and “to be” as recited are not believed to make the scope of the claims unascertainable in view of the context. However, the claims have been amended in a way believed to employ more definite language. Reconsideration is respectfully requested.

The Examiner further stated that the claims omitted a step. namely, the Examiner stated that the steps that occur when the object is not invalidated in the storage elements is needed. The Applicant disagrees. If the object is not invalidated, then the memory

remains as it was. There is no need to recite such a step. Notwithstanding the above, claims 1, 10, 11, 20 and 21 have been amended in a way believed to overcome the rejections. reconsideration is earnestly solicited.

By the Office Action, claims 1, 10, 11, 20 and 21 stand rejected under 35 U.S.C. §101, as being directed to non-patentable subject matter for lack of a tangible result.

The Applicant respectfully disagrees. However, claims 1, 10, 11, 20 and 21 have been amended to further clarify the invention by including updating the object after the invalidation time has passed to maintain consistency for the object across the plurality of storage elements by allowing the storage elements to serve an old version of the object until all active storage elements are ready to serve a new updated version of the object.

The tangible result provided includes at least that data remains consistent throughout a system. It is respectfully submitted that such a recitation is sufficient to overcome the rejection especially in view of the other features of the claims. Reconsideration of the rejection is earnestly solicited.

By the Office Action, claims 1-6, 9-16, 19-26 and 28 stand rejected under 35 U.S.C. §103 (a) as being unpatentable over U.S. Patent No. 5,528,511 to Hasegawa (hereinafter Hasegawa) in view of U.S. Patent Application No. 2002/0194382 to Kausik et al. (hereinafter Kausik).

The Applicant respectfully disagrees with the rejection.

Hasegawa is directed to a system that verifies delay times in logic circuits. If a signal input to a logic circuit is not properly timed the logic circuit may yield an incorrect result. The system of Hasegawa provides verification that the delay time is correct to ensure that the logic circuit operates correctly. While Hasegawa includes an invalidness specifier 20, this module specifies which of the rise or fall transitions is valid so that a verifier can assess the correct delay time. The Examiner stated that Hasegawa teaches determining delay times and delaying updating the object until the invalidation time has passed. However, the delay times taught by Hasegawa are not invalidation times for programming objects as recited in the present claims. Further, Hasegawa does not invalidate objects after an invalidation time has passed. Delay time verification for logic circuits is a completely different in scope, application and meaning from the teachings of the present claims. One skilled in the art would not look to the teachings of Hasegawa to understand how to maintain data consistency in a distributed memory system.

The Examiner further stated that Hasegawa fails to explicitly teach storing objects in a plurality of storage elements. This is consistent with the above discussion. Hasegawa is directed to a completely different art and does not even suggest the present elements of the claims. The Examiner cited Kausik to cure the deficiencies of Hasegawa.

Kausik is directed to a system for management of embedded objects. Kausik seeks to reduce network delays by eliminating the need to validate the status of an object at the cache device storing the object. The object is sent first to a proxy, which updates the header with an extended cache life. The proxy then writes the object to cache. Because of the extended life assigned to the object, the object remains in cache for a long time and does not

require validation during that period. The concepts presented by Kausik provide that a requested object is first modified to increase its lifetime to prevent a validation step. Kausik applies an arbitrary expiration time using a proxy to artificially trick the objects to prevent the objects from requesting validation from a content server. The proxy reduces network traffic to the server but the proxy provides other problems. For example, the proxy updates the URL of each object when the proxy assigns a new expiry time to the object. There is not teaching or suggestion in Kausik that the expiry time be a time selected such that each of the plurality of storage elements has either received notification of an update pursuant to the request or has declared itself failed. Instead, Kausik relies on a preassigned “maxage” [0019].

In addition, Kausik fails to disclose or suggest allowing the storage elements to serve an old version of the object until all active storage elements are ready to serve a new version of the object. In Kausik, each object is checked against the expiry time upon access. Only then is the object updated with the content server or proxy. At any given time multiple versions of the object can therefore exist. Nowhere in Kausik is it disclosed or suggested that the storage elements be allowed to serve an old version of the object until all active storage elements are ready to serve a new version of the object. In Kausik, new version are served before all of the old versions are invalidated.

The cited combination of Hasegawa and Kausik fails to disclose or suggest the present claims as amended. For example, claim 1 now recites, *inter alia*, storing an object in a plurality of storage elements; in response to a request to update the object, determining a future invalidation time when the object will be invalidated in storage elements currently storing the object, the invalidation time including a time selected such that each of the plurality of storage

elements has either received notification of an update pursuant to the request or has declared itself failed; delaying updating the object until the invalidation time has passed; and updating the object after the invalidation time has passed to maintain consistency for the object across the plurality of storage elements by allowing the storage elements to serve an old version of the object until all active storage elements are ready to serve a new version of the object.

The Applicant believes that there is no motivation to combine Hasegawa and Kausik, since Hasegawa is directed to logic circuit checking and Kausik is directed to a proxy for assigning expiry times to objects. However, even if combined, Hasegawa and/or Kausik, taken alone or in combination, fail to disclose or suggest at least: the invalidation time including a time selected such that each of the plurality of storage elements has either received notification of an update pursuant to the request or has declared itself failed ... and updating the object after the invalidation time has passed to maintain consistency for the object across the plurality of storage elements by allowing the storage elements to serve an old version of the object until all active storage elements are ready to serve a new version of the object.

The cited combination does not recite at least these elements. Support for the amendments is found throughout the specification. See e.g., page 12, line 17 to page 13 line 10, page 24, line 8-15, as well as other portions of the specification and drawings. Claims 10, 11, 20 and 21 have been amended to include similar recitations as claim 1 and are therefore believed to be in condition for allowance as well.

The dependent claims are believed to be in condition for allowance due at least to their dependencies, and for other reasons as well. For example, as recited in claim 4, a step of sending, from the consistency coordinator to a storage element and/or from a storage element

to the consistency coordinator, heart beat messages to obtain availability information, is not disclosed or suggested by the cited combination. Other reasons exist for allowing other dependent claims as well. The Examiner is respectfully requested to reconsider the dependent claims. It is respectfully submitted that claims 1-28 are in condition for allowance for at least the reasons stated. Reconsideration of the rejection is earnestly solicited.

In view of the foregoing amendments and remarks, it is respectfully submitted that all the claims now pending in the application are in condition for allowance. Early and favorable reconsideration of the case is respectfully requested.

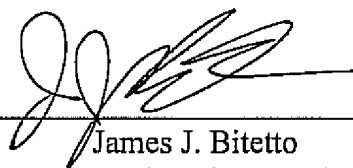
It is believed that no additional fees or charges are currently due. However, in the event that any additional fees or charges are required at this time in connection with the application, they may be charged to applicant's IBM Deposit Account No. 50-0510.

Respectfully submitted,

Date:

6/7/07

By:



James J. Bitetto
Registration No. 40,513

Mailing Address:

KEUSEY, TUTUNJIAN & BITETTO, P.C.
20 Crossways Park North, Suite 210
Woodbury, NY 11797
Tel: (516) 496-3868
Fax: (516) 496-3869